(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2013-0025660
(43) 공개일자 2013년03월12일

(51) 국제특허분류(Int. Cl.)
G06F 9/38 (2006.01)
(21) 출원번호 10-2011-0089095
(22) 출원일자 2011년09월02일
심사청구일자 2011년09월02일

(71) 출원인
고려대학교 산학협력단
서울 성북구 안암동5가 1
(72) 발명자
정연돈
서울특별시 중구 청계천로 400, 105동 812호 (황학동, 롯데캐슬 베네치아)
손지훈
서울특별시 도봉구 쌍문4동 성원아파트 102-603
최현식
서울특별시 성북구 인촌로16길 4-29 (안암동3가)
(74) 대리인
특허법인 신성

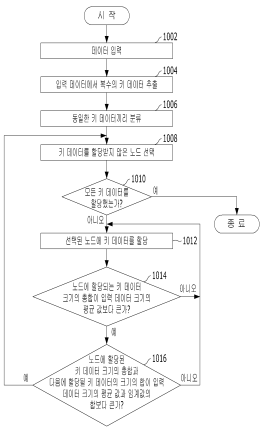
전체 청구항 수 : 총 6 항

(54) 발명의 명칭 **입력 데이터의 할당 방법 및 장치**

(57) 요 약

본 발명은 입력 데이터의 할당 방법 및 장치에 관한 것이다. 본 발명의 일 실시예에 따른 복수의 노드에 입력 데이터를 할당하는 방법은, 상기 입력 데이터에서 복수의 키 데이터를 추출하는 단계, 상기 입력 데이터에서 상기 복수의 키 데이터 각각이 차지하는 크기를 연산하는 단계, 상기 복수의 노드 중 상기 복수의 키 데이터를 할당받을 노드를 선택하는 단계 및 상기 선택된 노드에 할당되는 키 데이터 크기의 총합이 기준 크기를 초과하지 않을 때까지, 상기 선택된 노드에 상기 복수의 키 데이터를 할당하는 단계를 포함한다. 본 발명에 따르면 입력된 데이터를 복수의 노드에 보다 균등하게 할당함으로써, 대용량 데이터를 보다 빠른 시간 내에 효율적으로 처리할 수 있는 입력 데이터의 할당 방법 및 장치를 제공할 수 있는 효과가 있다.

대 표 도 - 도10



이 발명을 지원한 국가연구개발사업

과제고유번호	000438270110
부처명	중소기업청
연구사업명	산학연공동기술개발사업
연구과제명	(1차년도)L7 네트워크 트래픽 분석기 개발
주관기관	고려대학교 산학협력단
연구기간	2010.06.01 ~ 2011.05.31

특허청구의 범위

청구항 1

복수의 노드에 입력 데이터를 할당하는 방법에 있어서,

상기 입력 데이터에서 복수의 키 데이터를 추출하는 단계;

상기 입력 데이터에서 상기 복수의 키 데이터 각각이 차지하는 크기를 연산하는 단계;

상기 복수의 노드 중 상기 복수의 키 데이터를 할당받을 노드를 선택하는 단계; 및

상기 선택된 노드에 할당되는 키 데이터 크기의 총합이 기준 크기를 초과하지 않을 때까지, 상기 선택된 노드에 상기 복수의 키 데이터를 할당하는 단계를

포함하는 입력 데이터의 할당 방법.

청구항 2

제1 항에 있어서,

상기 기준 크기는 상기 입력 데이터 크기의 평균 값을 포함하고,

상기 평균 값은 상기 입력 데이터 크기의 총합을 노드의 수로 나눈 값인

입력 데이터의 할당 방법.

청구항 3

제1 항에 있어서,

상기 기준 크기는 상기 입력 데이터 크기의 임계 값을 포함하고,

상기 임계 값은 상기 평균 값에 총 노드의 수와 총 노드보다 1이 작은 수의 비를 곱한 값인

입력 데이터의 할당 방법.

청구항 4

복수의 노드에 입력 데이터를 할당하는 장치에 있어서,

상기 입력 데이터를 복수의 키 데이터로 분류하는 분류부;

상기 입력 데이터에서 상기 복수의 키 데이터 각각이 차지하는 크기를 연산하는 연산부; 및

상기 복수의 노드 중 상기 복수의 키 데이터를 할당받을 노드를 선택하고, 상기 선택된 노드에 할당되는 키 데이터 크기의 총합이 기준 크기를 초과하지 않을 때까지 상기 선택된 노드에 상기 복수의 키 데이터를 할당하는 할당부를

포함하는 입력 데이터의 할당 장치.

청구항 5

제4 항에 있어서,

상기 기준 크기는 상기 입력 데이터 크기의 평균 값을 포함하고,

상기 평균 값은 상기 입력 데이터 크기의 총합을 노드의 수로 나눈 값인

입력 데이터의 할당 장치.

청구항 6

제4 항에 있어서,

상기 기준 크기는 상기 입력 데이터 크기의 임계 값을 포함하고,

상기 임계 값은 상기 평균 값에 총 노드의 수와 총 노드보다 1이 작은 수의 비를 곱한 값인

입력 데이터의 할당 장치.

명 세 서

기 술 분 야

[0001] 본 발명은 입력 데이터의 할당 방법 및 장치에 관한 것으로, 특히 복수의 노드에 입력 데이터를 균등하게 할당하는 방법 및 장치에 관한 것이다.

배 경 기 술

[0002] 웹을 기반으로 한 정보 공유 및 교환이 활발해지고, 대용량 데이터를 한꺼번에 빠른 시간 내에 처리할 것이 요구되면서 하나의 서버로 데이터를 처리하는 것이 사실상 불가능해졌다. 이를 해결하기 위하여, 여러 대의 컴퓨터를 하나의 시스템으로 보아 데이터를 분배하는 논리 체계가 등장하게 되는데, 대표적인 것 중 하나가 맵리듀스(MapReduce) 기법이다.

[0003] 맵리듀스는 대용량 데이터를 여러 개의 노드 예컨대, 컴퓨터에 분배하여 동시에 처리하게 하는 기법으로서, 맵 단계 및 리듀스 단계로 이루어진다. 맵 단계 및 리듀스 단계는 각각 복수의 노드에서 이루어지며, 맵 단계에서는 데이터를 입력된 순서대로 처리하면서 하나의 데이터마다 고유의 식별 기호를 붙여 출력하고, 리듀스 단계에서는 맵 단계에서 출력된 고유 식별 기호가 붙은 데이터를 입력 받되, 고유 식별 기호가 동일한 데이터는 하나의 리듀스 노드의 입력으로 할당받는다.

[0004] 선 출원된 국내 특허 출원 "태스크 분배 및 병렬 처리 시스템과 그 방법"(2007.12.17. 10-2007-0132585)에는 맵 태스크 실행기에서 생성한 중간 결과 파일을 리듀스 태스크 실행기로 전달하는 시스템 및 방법이 개시되어 있고, "분산 병렬 처리 시스템의 다중 맵 태스크 중간 결과 정렬 및 결합 장치, 및 방법"(2007.12.18. 10-2007-0133589)에는 맵 함수를 적용하여 추출한 하나 이상의 키/값 쌍에 리듀스 태스크를 적용하여 중복 키를 제거한 중간 결과를 생성하는 방법이 개시되어 있다.

[0005] 다만, 종래의 맵리듀스 기법에 의하면, 리듀스 단계에서 복수의 노드에 할당되는 데이터의 크기가 각각 불균등한 경우에는, 대용량 데이터를 복수의 노드에서 한꺼번에 처리한다고 하더라도 총 입력 데이터 처리 시간은, 할당된 데이터의 크기가 가장 큰 노드가 데이터를 처리하는데 걸리는 시간이 되므로 효율이 반감되는 문제점이 있다. 이는, 리듀스 단계에서 노드에 입력되는 데이터의 고유 식별 기호의 수를 균등하게 할당한다 하더라도, 동일한 고유 식별 기호를 가진 데이터의 수가 각각 다르기 때문이다.

발명의 내용

해결하려는 과제

[0006] 본 발명은 입력된 데이터를 복수의 노드에 보다 균등하게 할당함으로써, 대용량 데이터를 보다 빠른 시간 내에 효율적으로 처리할 수 있는 입력 데이터의 할당 방법 및 장치를 제공하는 것을 일 목적으로 한다.

[0007] 본 발명의 목적들은 이상에서 언급한 목적으로 제한되지 않으며, 언급되지 않은 본 발명의 다른 목적 및 장점들은 하기의 설명에 의해서 이해될 수 있고, 본 발명의 실시예에 의해 보다 분명하게 이해될 것이다. 또한, 본 발명의 목적 및 장점들은 특허 청구 범위에 나타낸 수단 및 그 조합에 의해 실현될 수 있음을 쉽게 알 수 있을 것

이다.

과제의 해결 수단

[0008] 이러한 목적을 달성하기 위한 본 발명은 복수의 노드에 입력 데이터를 할당하는 방법에 있어서, 상기 입력 데이터에서 복수의 키 데이터를 추출하는 단계, 상기 입력 데이터에서 상기 복수의 키 데이터 각각이 차지하는 크기를 연산하는 단계, 상기 복수의 노드 중 상기 복수의 키 데이터를 할당받을 노드를 선택하는 단계 및 상기 선택된 노드에 할당되는 키 데이터 크기의 총합이 기준 크기를 초과하지 않을 때까지, 상기 선택된 노드에 상기 복수의 키 데이터를 할당하는 단계를 포함하는 것을 일 특징으로 한다.

[0009] 또한 본 발명은 복수의 노드에 입력 데이터를 할당하는 장치에 있어서, 상기 입력 데이터를 복수의 키 데이터로 분류하는 분류부, 상기 입력 데이터에서 상기 복수의 키 데이터 각각이 차지하는 크기를 연산하는 연산부 및 상기 복수의 노드 중 상기 복수의 키 데이터를 할당받을 노드를 선택하고, 상기 선택된 노드에 할당되는 키 데이터 크기의 총합이 기준 크기를 초과하지 않을 때까지 상기 선택된 노드에 상기 복수의 키 데이터를 할당하는 할당부를 포함하는 것을 또 다른 특징으로 한다.

발명의 효과

[0010] 전술한 바와 같은 본 발명에 의하면, 입력된 데이터를 복수의 노드에 보다 균등하게 할당함으로써, 대용량 데이터를 보다 빠른 시간 내에 효율적으로 처리할 수 있는 입력 데이터의 할당 방법 및 장치를 제공할 수 있다.

도면의 간단한 설명

[0011] 도 1a 및 도 1b는 맵 리듀스 기법을 설명하기 위한 도면.

도 2a 및 도 2b는 맵 리듀스 기법에 의한 데이터 분배를 설명하기 위한 도면.

도 3a 및 도 3b는 그리디 알고리즘(Greedy algorithm)에 의한 데이터 분배 방식.

도 4는 본 발명의 일 실시예에 따른 입력 데이터의 할당 장치를 설명하기 위한 도면.

도 5는 본 발명의 일 실시예에 따른 입력 데이터의 할당 과정을 설명하기 위한 도면.

도 6a 내지 도 6c는 본 발명의 일 실시예에 따른 입력 데이터 할당 결과를 종래 기술에 따른 입력 데이터 할당 결과와 비교한 도면.

도 7a 및 도 7b는 본 발명의 일 실시예에 따른 입력 데이터 할당을 위하여 기준 크기를 설정하는 실시예를 설명하기 위한 도면.

도 8은 본 발명의 일 실시예에 따른 균등한 입력 데이터 할당을 설명하기 위한 알고리즘의 의사코드(Pseudocode)를 나타낸 도면.

도 9는 본 발명의 일 실시예에 따른 입력 데이터의 할당 방법의 흐름을 나타내는 흐름도.

도 10은 본 발명의 다른 실시예에 따른 입력 데이터의 할당 방법의 흐름을 나타내는 흐름도.

발명을 실시하기 위한 구체적인 내용

[0012] 전술한 목적, 특징 및 장점은 첨부된 도면을 참조하여 상세하게 후술되며, 이에 따라 본 발명이 속하는 기술분야에서 통상의 지식을 가진 자가 본 발명의 기술적 사상을 용이하게 실시할 수 있을 것이다. 본 발명을 설명함에 있어서 본 발명과 관련된 공지 기술에 대한 구체적인 설명이 본 발명의 요지를 불필요하게 흐릴 수 있다고 판단되는 경우에는 상세한 설명을 생략한다. 이하, 첨부된 도면을 참조하여 본 발명에 따른 바람직한 실시예를 상세히 설명하기로 한다. 도면에서 동일한 참조부호는 동일 또는 유사한 구성요소를 가리키는 것으로 사용된다.

[0013] 도 1a 및 도 1b는 맵 리듀스 기법을 설명하기 위한 도면을 나타낸다.

[0014] 도 1a에 나타난 바와 같이, 맵 리듀스 기법은 복수의 노드로 입력 데이터가 입력되면, 먼저 맵 단계에서 입력 데이터를 처리하여 중간 결과를 출력하고, 출력된 중간 결과를 리듀스 단계를 처리하는 복수의 노드에 분배하며, 리듀스 단계에서 입력받은 중간 결과를 처리하여 최종 결과를 출력한다. 이 과정을 통하여, 입력받은 대용량 데이터를 복수의 노드에 균등하게 분배하면 데이터 처리 속도가 빨라져 효율적인 데이터 처리가 가능해진다.

[0015] 도 1b를 참조하면, 맵리듀스 기법으로 문서 데이터를 처리하기 위하여 복수의 노드에 문서로 된 데이터를 나누어 입력하면, 맵퍼(Mapper)는 문서를 구성하는 각각의 단어를 처리하여 중간 결과를 출력한다. 즉, 단어 표기를 고유 식별 기호로 추출하고, 단어의 개수를 데이터의 크기로 추출하여, 고유 식별 기호 및 데이터의 크기를 한 쌍의 중간 데이터로 출력한다. 분배부(Partitioner)는 입력된 순차대로 처리된 중간 데이터를 입력받은 후, 표기가 동일한 단어 즉, 고유 식별 기호가 동일한 데이터를 분류하고, 고유 식별 기호가 동일한 데이터를 하나의 리듀서(Reducer)에 분배한다. 리듀서는 고유 식별 기호가 동일한 데이터의 크기를 연산하여 중복되는 데이터를 제거함으로써, 이후 단계에서 처리할 데이터의 수를 감소시킬 수 있다. 이하, 고유 식별 기호가 동일한 데이터를 키 데이터라 한다.

[0016] 도 2a 및 도 2b는 맵 리듀스 기법에 의한 데이터 분배를 설명하기 위한 도면이다.

[0017] 도 2a에 나타난 바와 같이, 도 1b의 맵퍼에서 출력된 복수의 중간 데이터는 복수의 리듀서에 분배되는데, 리듀서 즉, 노드의 수가 데이터의 수보다 적은 경우가 일반적이므로, 한정된 리듀서에 중간 데이터를 분배하는 과정이 필요하다.

[0018] 도 2b를 참조하면, 막대에 표기된 숫자는 고유 식별 기호를 나타내고, 막대의 길이는 데이터의 크기를 나타낸다. 각 노드에 분배하는 고유 식별 기호의 개수가 균등해지면, 각 고유 식별 기호를 가지는 데이터의 크기가 불균등해지므로 효율적인 데이터 처리가 어려워진다. 따라서, 각 노드에 분배하는 고유 식별 기호의 개수를 달리하여, 각각의 리듀서에 보다 균등한 크기의 데이터를 분배할 수 있다.

[0019] 도 3a 및 도 3b는 그리디 알고리즘(Greedy algorithm)에 의한 데이터 분배 방식을 나타낸다.

[0020] 도 3a 및 도 3b에 나타난 바와 같이, 그리디 알고리즘에 따르면, 각 리듀서에 균등한 크기의 데이터를 분배하기 위하여, 맵퍼에서 출력된 중간 데이터를 데이터 크기가 큰 순서대로 정렬(Sort)하고, 크기 순으로 정렬된 데이터를 차례대로 노드에 분배한다.

[0021] 이때, 각 노드에 할당되는 키 데이터의 모든 키의 종류를 메모리에 저장할 수 있다. 즉, 메모리에 저장된 키의 종류를 바탕으로 선택된 노드에 할당된 데이터의 크기를 알 수 있고, 할당된 데이터의 크기에 기반하여 다음 데이터를 처리할 리듀서를 선택한다. 예컨대, 도 3b의 4개의 노드에 데이터 크기가 큰 순서대로 중간 데이터를 분배하고, 각 노드마다 분배된 키의 종류에 대한 정보가 메모리에 저장되면 이를 이용하여, 분배된 데이터 크기가 작은 노드부터 다음 중간 데이터를 분배한다. 분배되는 중간 데이터의 크기는 점점 작아지므로, 각 노드에 균등한 크기의 데이터를 분배할 수 있을 것이다. 다만, 입력되는 데이터의 양이 증가할수록 즉, 고유 식별 기호의 개수가 증가할수록 메모리 사용량이 증가하므로 처리 가능한 데이터의 양에 한계가 있다.

[0022] 도 4는 본 발명의 일 실시예에 따른 입력 데이터의 할당 장치를 설명하기 위한 도면이다.

[0023] 도 4를 참조하면, 복수의 노드에 입력 데이터를 할당하는 장치(402)는, 분류부(404), 연산부(406) 및 할당부(408)를 포함한다.

[0024] 분류부(404)는 입력 데이터를 복수의 키 데이터로 분류한다. 키 데이터란, 대용량의 입력 데이터를 각각 하나의 데이터로 분류하면서 키를 붙여 구별하는 것으로써, 키는 일종의 고유 식별 기호의 역할을 할 수 있다. 이때, 키 데이터는 입력 데이터가 분류되는 순서대로 정렬될 수 있다.

[0025] 연산부(406)는 입력 데이터에서 복수의 키 데이터 각각이 차지하는 크기를 연산한다. 즉, 복수의 키 데이터 중 동일한 키를 가지는 데이터를 그루핑(Grouping)함으로써, 키 데이터 각각이 입력 데이터 전체에서 차지하는 크

기를 연산할 수 있다.

[0026] 할당부(408)는 복수의 노드 중 복수의 키 데이터를 할당받을 노드를 선택하고, 선택된 노드에 할당되는 키 데이터 크기의 총합이 기준 크기를 초과하지 않을 때까지 선택된 노드에 복수의 키 데이터를 할당한다.

[0027] 한편, 기준 크기는 입력 데이터의 평균 값으로써, 평균 값은 입력 데이터 크기의 총합을 노드의 수로 나눈 값이다. 또는, 기준 크기는 입력 데이터 크기의 임계 값으로써, 임계 값은 평균 값에 총 노드의 수와 총 노드보다 1이 작은 수의 비를 곱한 값이다. 또 다른 실시예에서, 기준 크기는 평균 값과 임계 값의 합으로 계산할 수 있다. 이 경우, 선택된 노드에 데이터를 계속 할당하되, 하나의 노드에 할당된 데이터의 크기가 기준 크기를 초과하지 않는 크기까지 키 데이터를 할당할 수 있다. 이때, 정렬된 순서대로 할당되는 키 데이터의 키의 종류가 메모리에 저장될 수 있고, 메모리에 저장되는 키의 종류는 선택된 노드에 최초로 할당된 키의 종류와 최후에 할당된 키의 종류를 포함할 수 있다. 이와 같이 메모리를 통하여, 선택된 노드에 할당된 모든 키의 종류가 아닌, 정렬 순서대로 할당된 최초 및 최후의 키 데이터의 키의 종류만을 저장함으로써, 입력 데이터의 양이 증가하더라도 메모리를 적게 사용할 수 있다.

[0028] 도 5는 본 발명의 일 실시예에 따른 입력 데이터의 할당 과정을 설명하기 위한 도면이다.

[0029] 본 발명의 일 실시예로서, 기준 크기는 입력 데이터 크기의 평균 값과 임계 값의 합으로 계산할 수 있다.

[0030] 도 5를 참조하면, 본 발명의 일 실시예로서, 복수의 노드에 복수의 중간 데이터를 할당함에 있어서, 먼저, 데이터를 할당받을 노드를 선택하고(노드1), 키 데이터가 분류된 순서대로 노드 1에 키 데이터를 계속 할당한다. 키 데이터 1,2가 할당된 노드 1에 키 데이터 3을 할당하면, 노드 1에 할당된 키 데이터의 총합이 기준 크기를 초과하므로, 키 데이터 3은 노드 1에 할당될 수 없고, 키 데이터 3이 할당될 새로운 노드를 선택하여야 한다. 이와 동일한 과정을 모든 입력 데이터를 할당할 때까지 반복하면, 복수의 노드에 키 데이터를 균등하게 할당할 수 있다.

[0031] 도 6a 내지 도 6c는 본 발명의 일 실시예에 따른 입력 데이터 할당 결과를 종래 기술에 따른 입력 데이터 할당 결과와 비교한 도면이다.

[0032] 종래 기술에 따른 입력 데이터 할당 결과는 HKD(Hash Key Distribution)로 나타낼 수 있고, 본 발명의 일 실시예에 따른 입력 데이터 할당 결과는 SKD(Skew-tolerant Key Distribution)로 나타낼 수 있다. 이상적인 할당 결과는 OPT로 나타낼 수 있다.

[0033] 도 6a를 참조하면, HKD에 의하면 복수의 노드 즉, 리듀서에 분배된 키는 동일하나, 복수의 노드에 분배된 데이터의 크기는 불균등하다. 반면에, SKD에 의하면 복수의 노드에 분배된 키의 수를 불균등하게 함으로써, 복수의 노드에 분배된 데이터의 크기를 균등하게 조절할 수 있다. 데이터의 크기를 나타낸 그래프(b)를 살펴보면, SKD 즉, 본 발명의 일 실시예에 따른 결과가 이상적인 결과인 OPT와 일치한다.

[0034] 도 6b는 HKD에 의하여 입력 데이터를 처리하는데 소요되는 시간을, 도 6c는 SKD에 의하여 입력 데이터를 처리하는데 소요되는 시간을 그래프로 나타낸 도면이다. 맵 단계에서 소요되는 시간은 동일하나, 리듀스 단계에서 소요되는 시간은 SKD에 의할 때 더 적어진다. 즉, 리듀서에 분배된 데이터의 크기가 균등하여, 분배된 데이터의 크기가 가장 큰 노드가 데이터를 처리하는데 걸리는 시간이 줄어들 수 있다.

[0035] 도 7a 및 도 7b는 본 발명의 일 실시예에 따른 입력 데이터 할당을 위하여 기준 크기를 설정하는 실시예를 설명하기 위한 도면이다.

[0036] 임계 값(도 7a의 Θ)은, 선택된 노드에 할당되는 마지막 키 데이터가, 선택된 그 노드에 할당될 것인가 아닌가를 판단하는 기준이 될 수 있다. 이때, 임계 값이 너무 작으면, 대부분의 노드에는 평균 값보다 작은 크기의 데이터가 할당되고, 마지막으로 선택되는 노드에만 평균 값보다 큰 크기의 데이터가 할당될 것이다. 반면에, 임계 값이 너무 크면, 대부분의 노드에 평균 값보다 큰 크기의 데이터가 할당되고, 마지막으로 선택되는 노드에는 평균 값보다 작은 크기의 데이터가 할당될 것이다.

[0037] 도 7a를 참조하면, 최악의 경우, 각각의 노드에 할당되지 못하는 다음 키 데이터는 크기가 가장 큰 키 데이터

(d_{max})가 될 것이고, 가장 큰 키 데이터는 마지막에 선택되는 노드에 할당될 수 있다.

[0038] 도 7b를 참조하면, 최악의 경우, 가장 큰 키 데이터가 할당되지 않은 각각의 노드에는 평균 값(mean)과 임계 값의 합에 가장 큰 키 데이터를 뺀 크기의 데이터($D_{mean} + D_\Theta - d_{max}$)가 할당될 것이다. 마지막으로 선택되는 노드(N_m)에 할당되는 데이터 크기를 D_m이라 할 때, $D_m + (m - 1) \times (D_{mean} - (d_{max} - D_\Theta)) = m \times D_{mean}$이므로, $D_m = D_{mean} + (m - 1) \times (d_{max} - D_\Theta)$가 된다. 이때, 모든 노드에 모든 데이터가 기준 크기 이하로 분배되어야 하므로, $D_m \leq D_{mean} + D_\Theta$를 만족하여야 한다. 따라서, $D_{mean} + (m - 1) \times (d_{max} - D_\Theta) \leq D_{mean} + D_\Theta$로부터 D_Θ를 계산할 수 있다.

[0039] 즉, 최악의 경우라면, $$D_\Theta \geq \frac{m-1}{m} \times d_{max}$$ 를 만족하여야 한다.

[0040] 다만, 일반적으로, 각각의 노드에 할당되지 못하는 다음 키 데이터는 기대 값인 E(d)를 가진다고 가정할 때, E(d)는 입력 데이터의 총합을 노드의 수로 나눈 수로서, $$E(d) = \frac{\sum_{i=1}^{n} di}{n}$$ 로 나타낼 수 있다.

[0041] 따라서, 임계 값은 $$D_\Theta = \frac{m-1}{m} \times E(d)$$ 로 계산할 수 있다.

[0042] 도 8은 본 발명의 일 실시예에 따른 균등한 입력 데이터 할당을 설명하기 위한 알고리즘의 의사코드(Pseudocode)를 나타낸 도면이다.

[0043] 도 8을 참조하면, 평균 값에 이르기까지 하나의 노드에 중간 데이터를 할당하고, 마지막에 할당되는 데이터로 인하여 노드에 할당된 데이터의 총합이 평균 값과 임계 값의 합의 미만이면, 데이터를 더 할당할 수 있으나, 초과하는 경우 다른 노드를 선택하는 과정을 의사코드로 표현하였다.

[0044] 도 9는 본 발명의 일 실시예에 따른 입력 데이터의 할당 방법의 흐름을 나타내는 흐름도이다.

[0045] 도 9를 참조하면, 먼저, 복수의 노드에 할당된 입력 데이터에서 복수의 키 데이터를 추출한다(902). 입력 데이터에서 추출한 복수의 키 데이터 각각이 차지하는 크기를 연산한다(904). 이후, 복수의 노드 중 복수의 키 데이터를 할당받을 노드를 선택하고(906), 선택된 노드에 할당되는 키 데이터 크기의 총합이 기준 크기를 초과하지 않을 때까지 선택된 노드에 복수의 키 데이터를 할당한다(908).

[0046] 이때, 기준 크기로서, 입력 데이터 크기의 평균 값을 포함할 수 있다. 평균 값은 입력 데이터 크기의 총합을 노드의 수로 나눈 값이다.

[0047] 또 다른 실시예에서, 기준 크기로서, 입력 데이터 크기의 임계 값을 포함할 수 있다. 임계 값은 평균 값에 총 노드의 수와 총 노드보다 1이 작은 수의 비를 곱한 값이다. 임계 값에 따라, 할당된 데이터의 총합이 가장 큰 노드에 할당된 데이터의 크기가 다른 노드에 할당된 데이터 크기와 균등해질 수 있다.

[0048] 도 10은 본 발명의 다른 실시예에 따른 입력 데이터의 할당 방법의 흐름을 나타내는 흐름도이다.

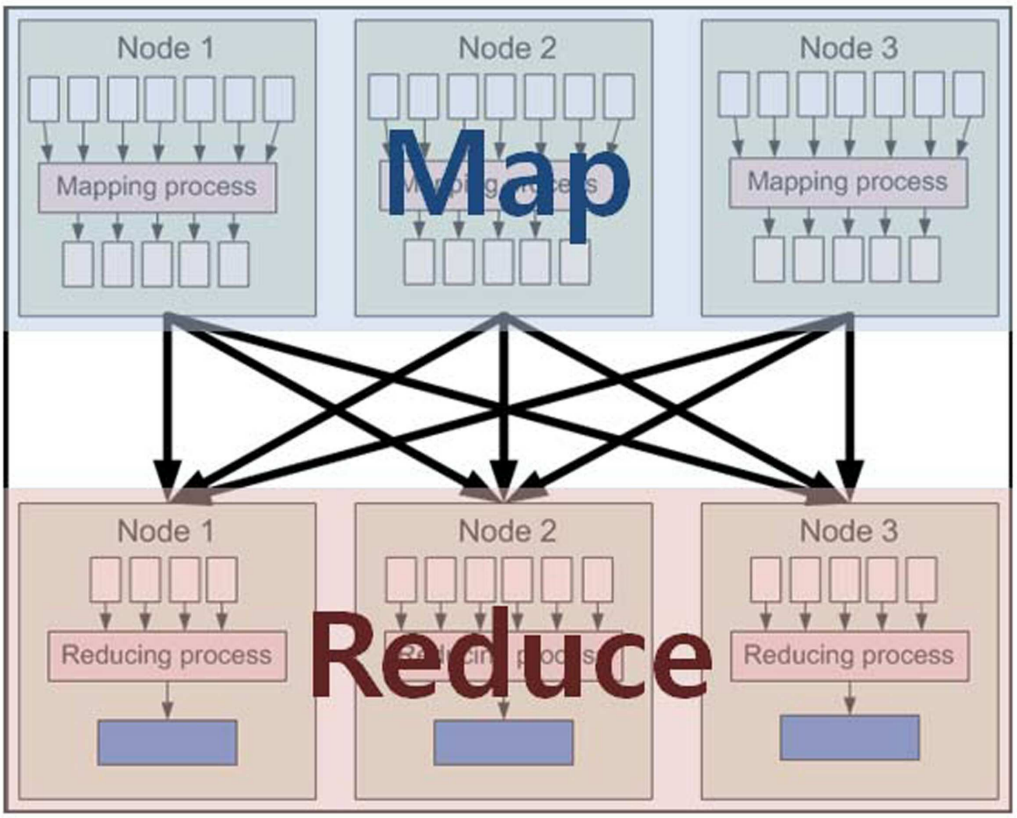
[0049] 도 10을 참조하면, 복수의 노드에 데이터가 입력되면(1002), 앞서 설명한 바와 같이, 입력 데이터에서 복수의 키 데이터를 추출한다(1004). 추출한 키 데이터 중 동일한 키 데이터끼리 분류(1006)한 후, 키 데이터를 할당받지 않은 노드를 선택한다(1008). 키 데이터가 모두 할당된 경우가 아니라, 할당하여야 할 키 데이터가 남아 있는 경우(1010)에는, 선택된 노드에 키 데이터를 할당하고(1012), 노드에 할당되는 키 데이터 크기의 총합이 입력 데이터 크기의 평균 값보다 이하이면 계속해서 선택된 노드에 키 데이터를 할당하고(1012), 평균 값을 초과하면 다음 단계로 넘어간다(1014). 다음 단계(1016)는, 노드에 할당되는 키 데이터 크기의 총합과 다음에 할당

될 키 데이터 크기의 합이 입력 데이터 크기의 평균 값과 임계 값의 합을 초과하는 경우라면, 더이상 선택된 노드에 키 데이터를 할당하지 않고, 키 데이터가 할당되지 않은 다른 노드를 선택하고(1008), 평균 값과 임계 값의 합 미만인 경우라면 계속해서 키 데이터를 할당한다(1012). 이 과정은, 모든 키 데이터를 할당할 때까지 반복될 수 있고, 모든 키 데이터가 할당되면 종료될 수 있다(1010).

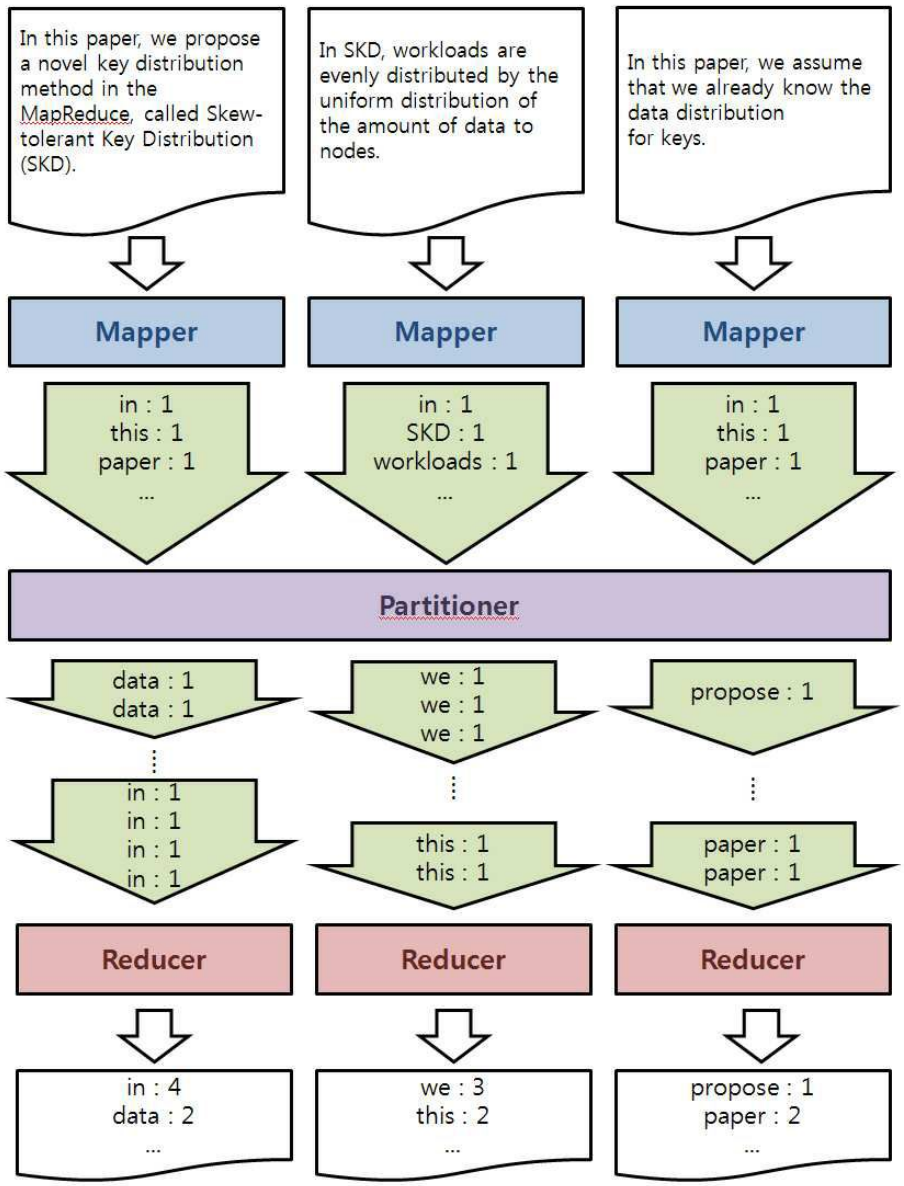
[0050] 전술한 본 발명은, 본 발명이 속하는 기술 분야에서 통상의 지식을 가진 자에게 있어 본 발명의 기술적 사상을 벗어나지 않는 범위 내에서 여러 가지 치환, 변형 및 변경이 가능하므로 전술한 실시예 및 첨부된 도면에 의해 한정되는 것이 아니다.

도면

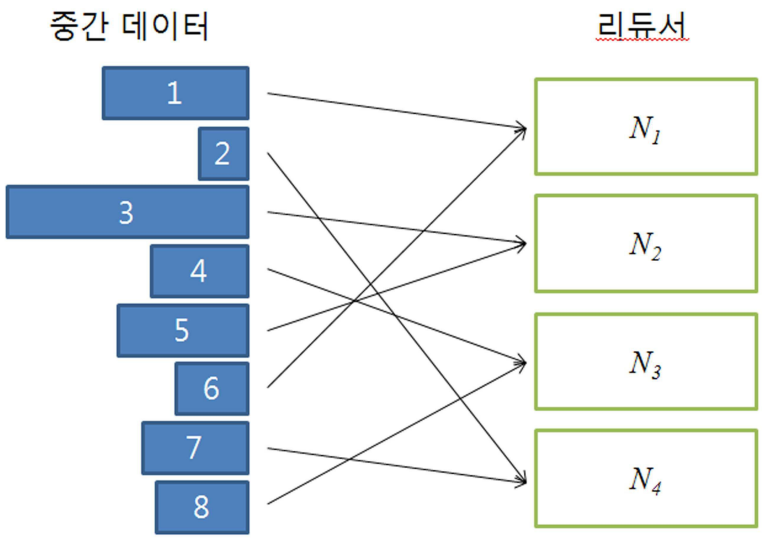
도면1a



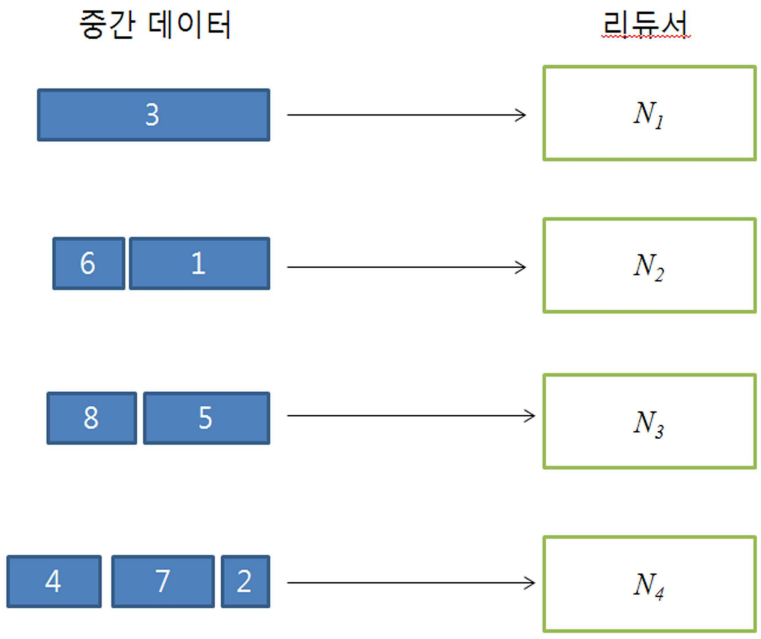
도면1b


In this paper, we propose a novel key distribution method in the MapReduce, called Skew-tolerant Key Distribution (SKD).
In SKD, workloads are evenly distributed by the uniform distribution of the amount of data to nodes.
In this paper, we assume that we already know the data distribution for keys.
Mapper
Mapper
Mapper
in : 1
this : 1
paper : 1
...
in : 1
SKD : 1
workloads : 1
...
in : 1
this : 1
paper : 1
...
Partitioner
data : 1
data : 1
we : 1
we : 1
we : 1
propose : 1
in : 1
in : 1
in : 1
in : 1
this : 1
this : 1
paper : 1
paper : 1
Reducer
Reducer
Reducer
in : 4
data : 2
...
we : 3
this : 2
...
propose : 1
paper : 2
...

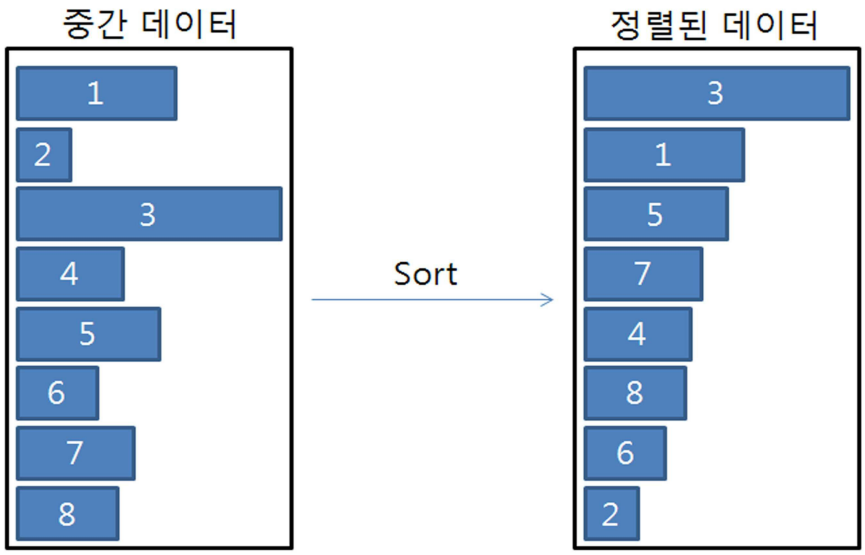
도면2a



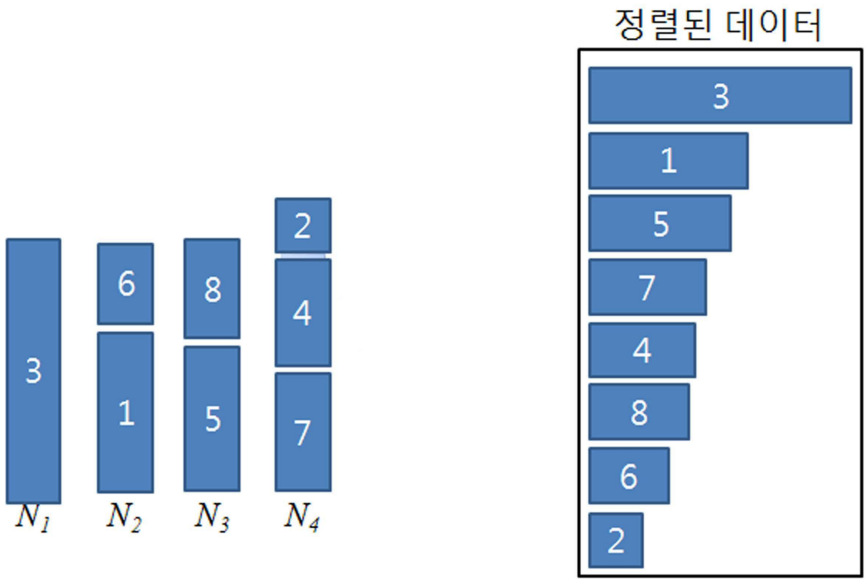
도면2b



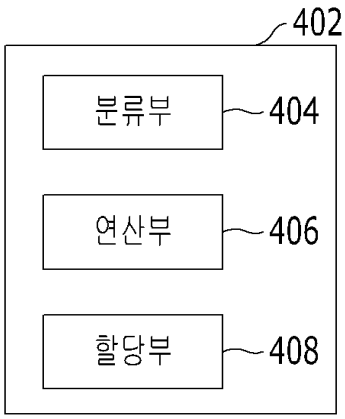
도면3a


중간 데이터
1
2
3
4
5
6
7
8
Sort
정렬된 데이터
3
1
5
7
4
8
6
2

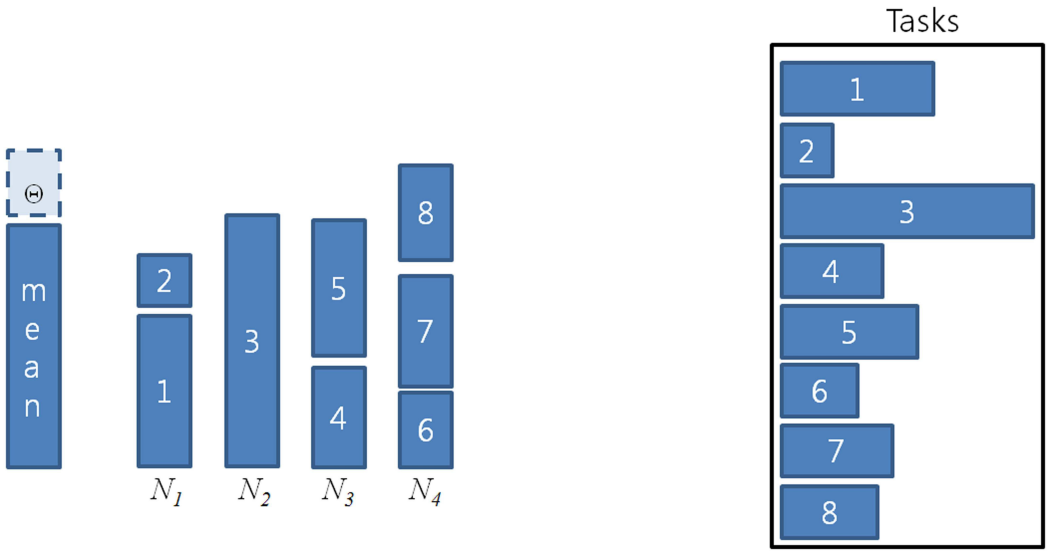
도면3b


3
6
1
8
5
2
4
7
N_1
N_2
N_3
N_4
정렬된 데이터
3
1
5
7
4
8
6
2

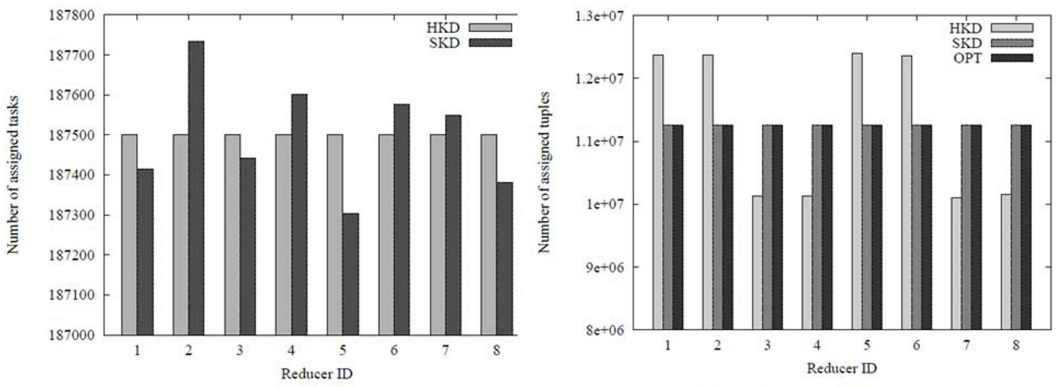
도면4



도면5



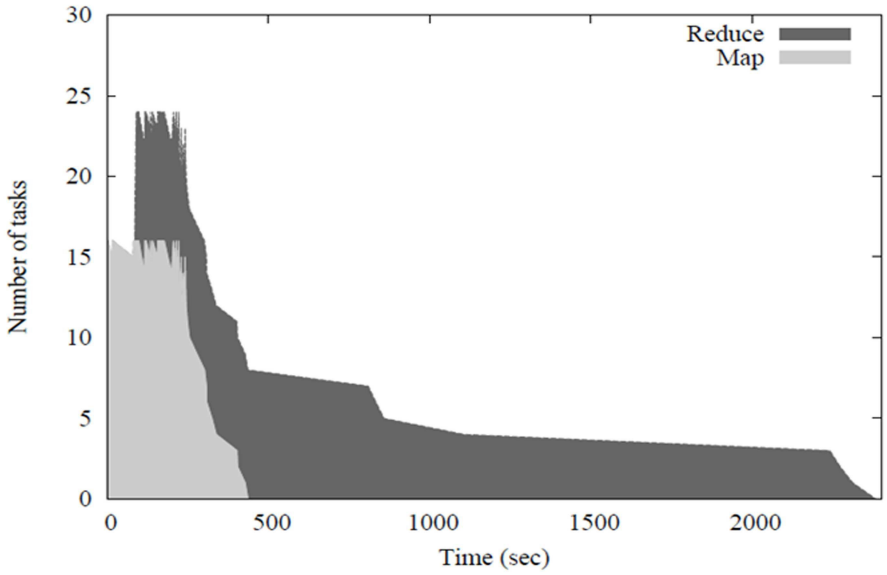
도면6a



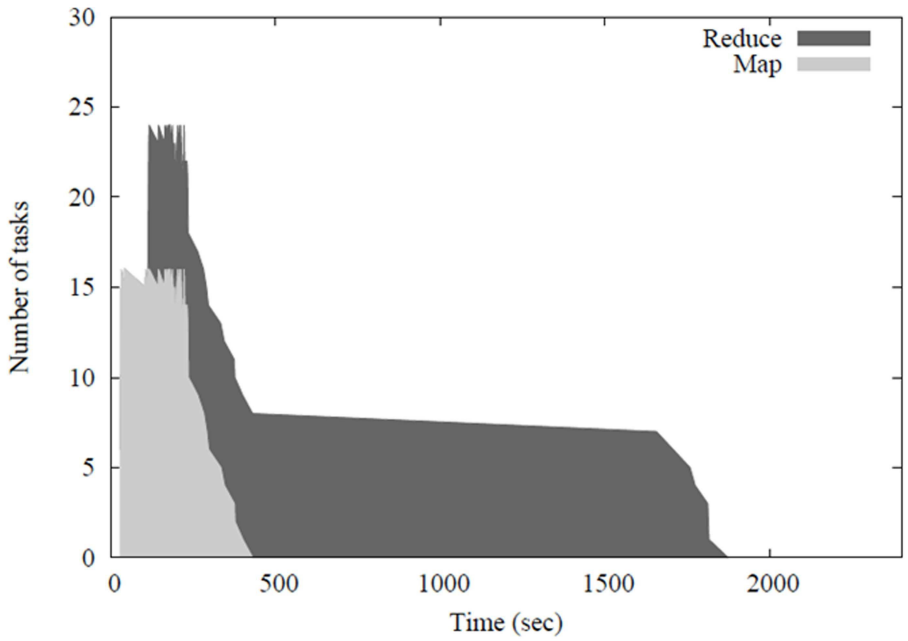
(a) Key assignment

(b) Data assignment

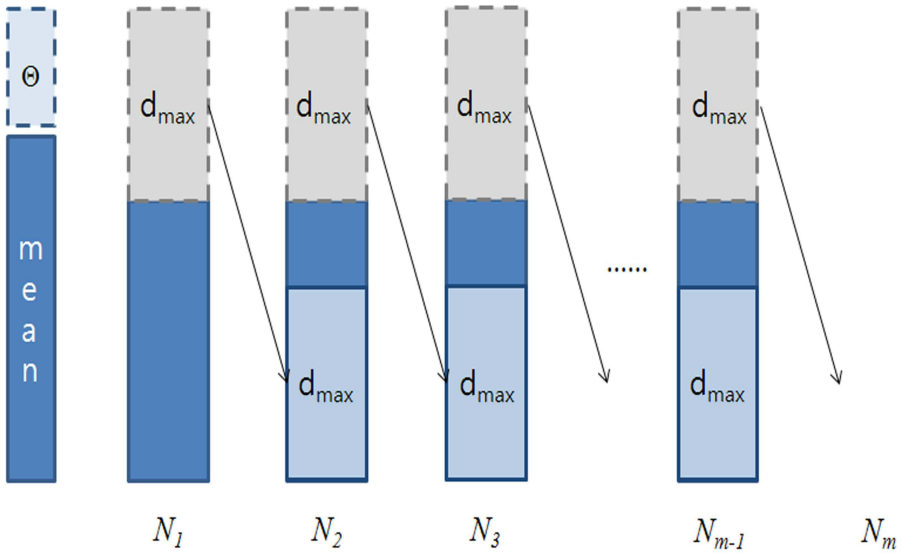
도면6b


Reduce
Map
Number of tasks
Time (sec)
0
5
10
15
20
25
30
0
500
1000
1500
2000

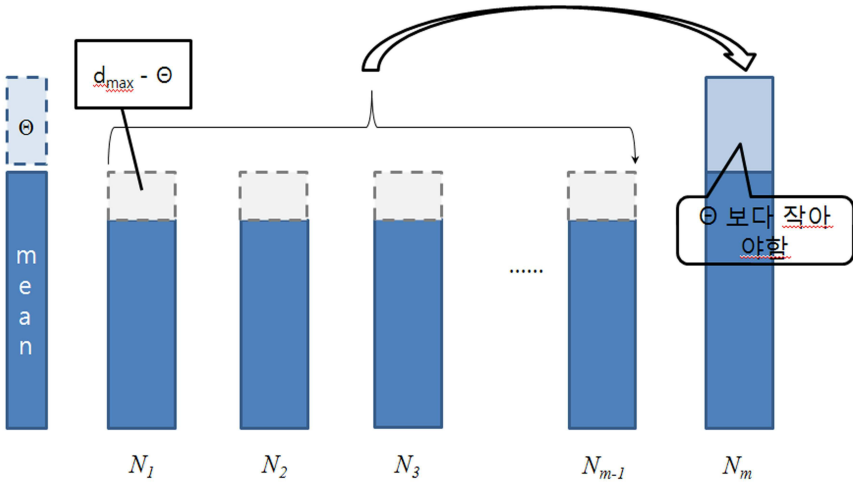
도면6c


Reduce
Map
Number of tasks
Time (sec)
0
5
10
15
20
25
30
0
500
1000
1500
2000

도면7a


Θ
mean
d_{max}
d_{max}
d_{max}
d_{max}
d_{max}
d_{max}
d_{max}
.....
N_1
N_2
N_3
N_{m-1}
N_m

도면7b

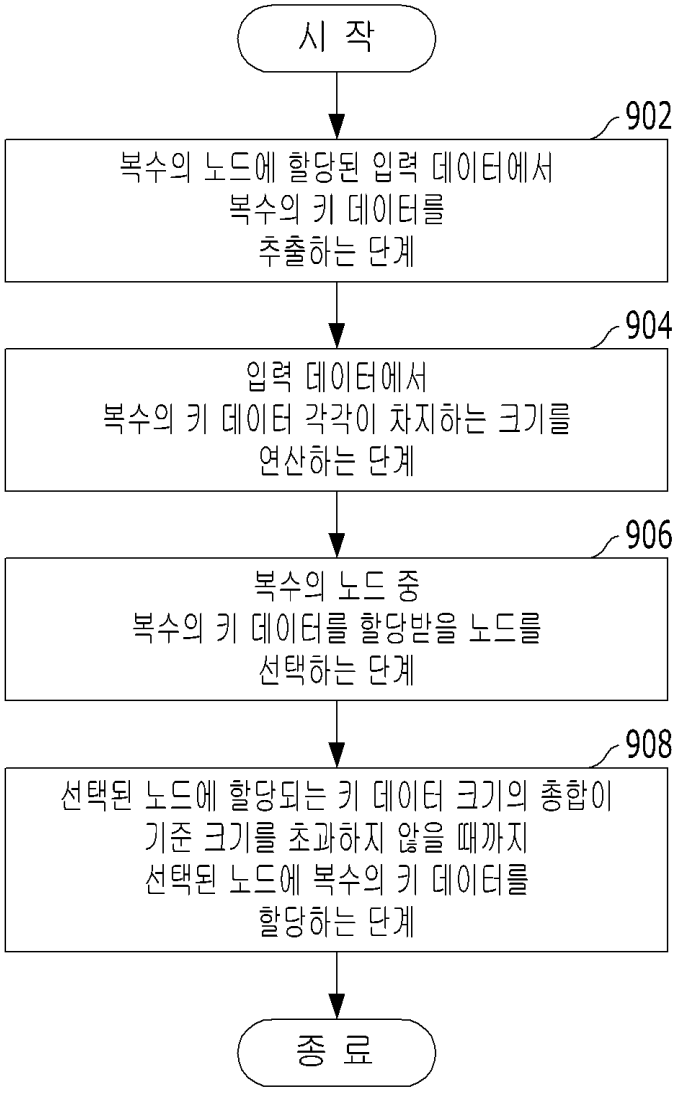

d_{max} - Θ
Θ
mean
Θ 보다 작아 야함
.....
N_1
N_2
N_3
N_{m-1}
N_m

도면8

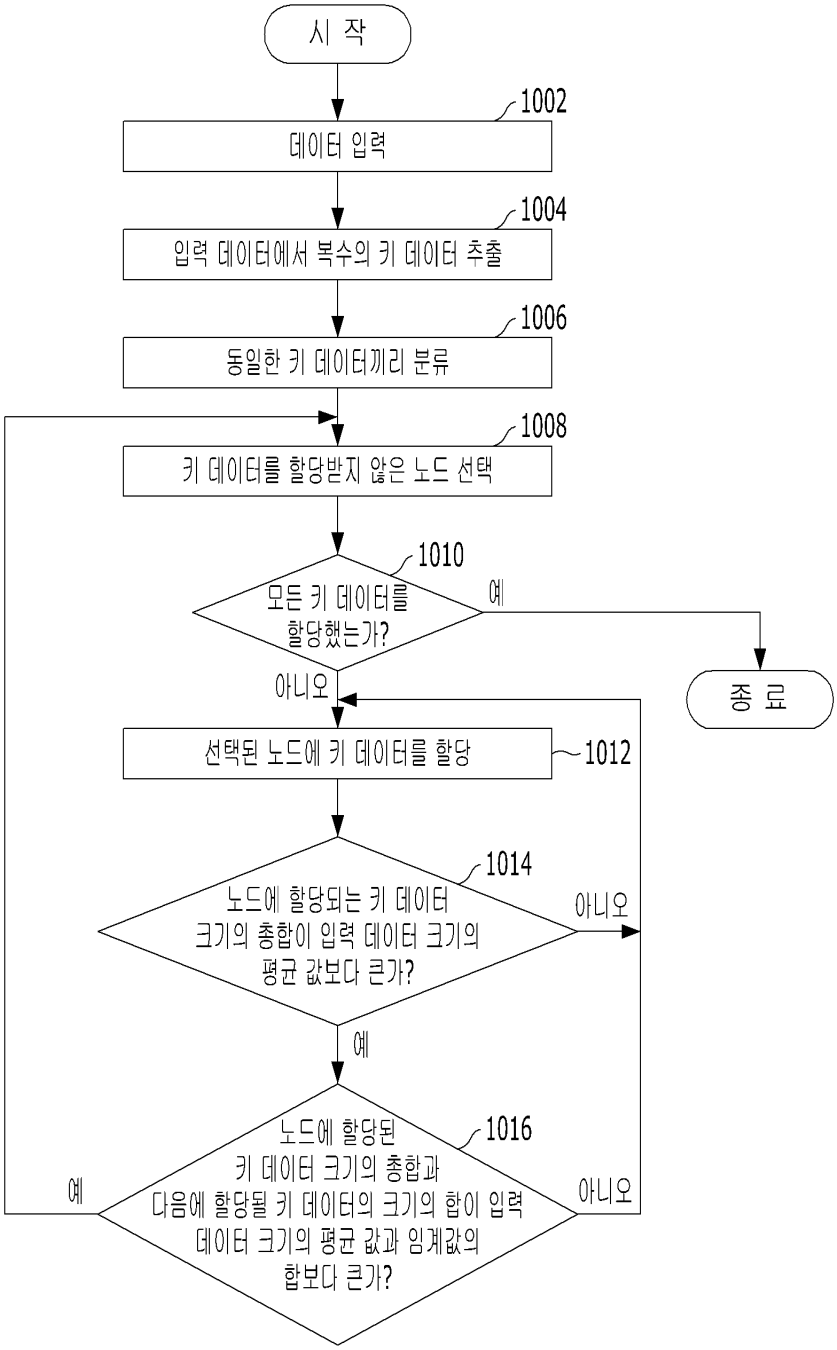
Algorithm 1 Balanced key assignment algorithm

```
1:  procedure BALANCEDKEYASSIGNMENT(a set of keys K, a set of data D, D_mean, D_θ)
2:      Sort K by IDs.
3:      A ← φ                                          ▷ A is an assignment.
4:      j ← 1
5:      while j ≤ n do
6:          k_j ∈ K
7:          for each node N_i ∈ N do
8:              K_i ← φ
9:              while D_i ≤ D_mean do
10:                 K_i ← K_i ∪ k_j
11:                 D_i ← D_i + d_j
12:                 j ← j + 1
13:             end while
14:             if D_i + d_j ≤ D_mean + D_θ then
15:                 K_i ← K_i ∪ k_j
16:                 D_i ← D_i + d_j
17:                 j ← j + 1
18:             end if
19:             A ← A ∪ K_i
20:         end for
21:     end while
22:     return A
23: end procedure
```

도면9


시 작
902
복수의 노드에 할당된 입력 데이터에서
복수의 키 데이터를
추출하는 단계
904
입력 데이터에서
복수의 키 데이터 각각이 차지하는 크기를
연산하는 단계
906
복수의 노드 중
복수의 키 데이터를 할당받을 노드를
선택하는 단계
908
선택된 노드에 할당되는 키 데이터 크기의 총합이
기준 크기를 초과하지 않을 때까지
선택된 노드에 복수의 키 데이터를
할당하는 단계
종 료

도면10


시 작
1002
데이터 입력
1004
입력 데이터에서 복수의 키 데이터 추출
1006
동일한 키 데이터끼리 분류
1008
키 데이터를 할당받지 않은 노드 선택
1010
모든 키 데이터를 할당했는가?
예
종 료
아니오
선택된 노드에 키 데이터를 할당
1012
1014
노드에 할당되는 키 데이터 크기의 총합이 입력 데이터 크기의 평균 값보다 큰가?
아니오
예
1016
노드에 할당된 키 데이터 크기의 총합과 다음에 할당될 키 데이터의 크기의 합이 입력 데이터 크기의 평균 값과 임계값의 합보다 큰가?
예
아니오